Hybrid-Level Instruction Injection for Video Token Compression in Multi-modal Large Language Models

Supplementary Material

Overview. In the Supplementary Material, we introduce more implementation details in Appendix A, more details about the conditional pre-training stage and the constructed HICom-248K dataset in Appendix B. Then we add more experiments in Appendix C, including more ablation studies, more benchmarks, and more qualitative analysis.

A. Implementation Details

We use SigLIP [64] (so400m-patch14-384) as our vision encoder and text encoder, and choose Qwen2.5 series [48] as our LLMs. At the global-level, the number of the learnable queries is set to 32 referred to Qformer [22]. We mainly follow LLaVA-OneVision [20] to configure our training hyper-parameters and settings. The detailed configurations are shown in Tab. A1. Different from LLaVA-OneVision, we keep our vision encoder frozen at all stages. Compared with the *Stage1.5* of LLaVA-OneVision, the training goal of our conditional pre-training stage is quite different. Therefore, we use a few different training strategies. Specifically, we keep the LLM frozen at the conditional pre-training stage as it is designed to align the compressing module based on the instruction condition. We also use a higher learning rate at this stage, 1e-3 for the parameters of instruction injection, and 1e-4 for other parameters in the compressing module.

We use 3D position embedding for our global-level instruction injection, as the input includes three dimensions: time, width, and height. Following CogVideoX [62] and Qwen2VL [52], we extend 2D absolute position embedding to 3D. Each latent in the video tensor can be represented by a 3D coordinate. We occupy 3/8, 3/8, and 2/8 of the hidden states' channel of position embedding. The resulting encoding is then concatenated along the channel dimension to obtain the final 3D positional encoding.

Table A1. Detailed training configurations for each stage. We follow LLaVA-OneVision [20] to choose our configurations. At the conditional pre-training stage and instruction tuning stage, we use a global batch size of 512 for the 0.5B model, and 256 for the 7B and 72B models. Comp. denotes our compressing module, which plays the role of compressing the visual tokens and projecting them into the LLM's embedding space.

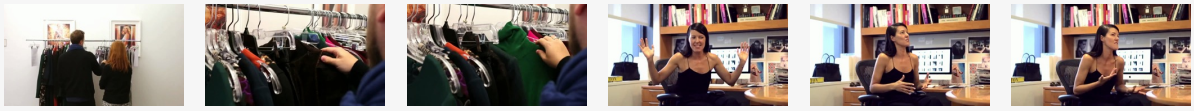
	Alignment	**Conditional Pre-train**	**Instruction Tuning**
Data	Image	Video	Video
# Tokens	81+32	648+32	648+32
# Samples	558K	248K	2.6M
Trainable	Comp.	Comp.	Comp., LLM
7B LLM	63M	63M	7.7B
Batch size	512	256/512	256/512
lr: Vision Enc.	-	-	-
lr: inj. In Comp.	-	1×10^{-3}	1×10^{-5}
lr: others in Comp.	1×10^{-3}	1×10^{-4}	1×10^{-5}
lr: LLM	-	-	1×10^{-5}
Epoch	1	1	1

B. New Dataset and Training Stage

B.1. Conditional Pre-training Stage

LLaVA [31] introduces a two-stage training pipeline for MLLMs, which first pre-trains for feature alignment and then conducts end-to-end instruction tuning. Mainstream methods typically adopt this two-stage training pipeline. During the alignment stage, image-caption pairs are commonly used to pre-train the visual projector, aligning visual features with the LLM's embedding space. At the instruction tuning stage, various types of question-answer pair data are utilized to fine-tune the model, including general QA, multiple-choice QA, OCR, documents/charts/screens, math reasoning, attribute perception, counting, temporal reasoning, information synthesis, *etc* [33], equipping the model with instruction-following capabilities. Recently, LLaVA-OneVision [20] proposes a three-stage training paradigm. Between *Language-Image Alignment* and *Visual Instruction Tuning*, it introduces a new *stage1.5*, named *High-Quality Knowledge Learning*, using re-captioned detailed description data, OCR data, and Chinese and language data, maintaining a training strategy similar to the instruction tuning stage to inject new knowledge into MLLMs.

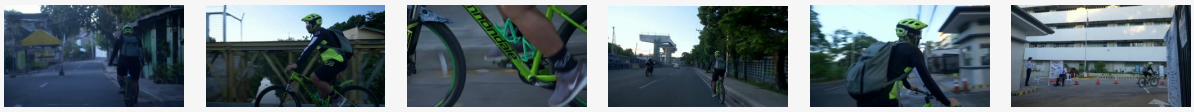
In our work, we propose a new training stage between alignment and instruction tuning. Unlike LLaVA-OneVision, our conditional pre-training stage is designed to pre-train the parameters for instruction injection. In the existing alignment stage, instructions typically prompt the model to describe visual content globally (e.g., "Please summarize the visual content of the image."), which is insufficient for selectively retaining information during token compression. Pre-training the instruction injection at this stage does not effectively teach the model to retain user-important information. By using conditional captions with appropriate instructions, we can pre-train the compression module to perform conditional alignment first, simplifying the instruction tuning for conditional compression. Thus,



Instruction: What are the two individuals doing in front of the clothing rack?
Description: The two individuals are examining various garments hanging on the clothing rack, possibly selecting or organizing clothes.

Instruction: How does the person seated at the desk interact with their surroundings?
Description: The person seated at the desk is gesturing with their hands while speaking, indicating they are engaged in some form of communication or explanation.

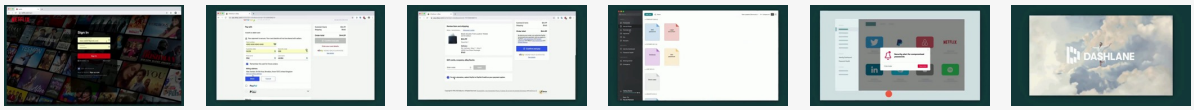
Instruction: What can be seen on the desk in the office-like setting?
Description: The desk in the office-like setting is cluttered with various items, including books, papers, and a computer monitor displaying images.



Instruction: What is the person riding?
Description: The person in the video is riding a bicycle.

Instruction: What is the person wearing while riding the bicycle?
Description: The person riding the bicycle is wearing a helmet and carrying a backpack.

Instruction: Where does the scene transition to after the person rides the bicycle along the street?
Description: The scene transitions from the outdoor street to an indoor or semi-enclosed space where the cyclist continues to ride.



Instruction: What is the purpose of the login screen shown in the video?
Description: The login screen displayed in the video is used for accessing Netflix, where users can enter their email address and password to sign in.

Instruction: What information is included on the eBay checkout page shown in the video?
Description: The eBay checkout page in the video contains payment details such as the card number, expiration date, and security code for completing a transaction.

Instruction: What does the dashboard interface in the video display?
Description: The dashboard interface in the video displays various icons representing different services like Twitter, LinkedIn, and PayPal, along with a security alert notification.

Figure A1. Some examples of our constructed HICom-248K instruction-following descriptions.

we introduce a new conditional pre-training stage utilizing our HICom-248K dataset, which implements conditional pre-training for conditional compression.

B.2. HICom-248K

HICom-248K dataset is designed for the conditional pre-training, which consists of video question-answer pairs. Since the goal of the conditional pre-training stage is to achieve conditional alignment based on the instruction, HICom-248K focuses on providing one type of data, *i.e.*, the instruction-followed descriptions, which meets the following requests:

- The instruction should refer to the specific information in the video, providing the guidance role of conditional compression.
- The answer should be the caption of the specific visual

Table A2. The pre-defined 29 categories during the collection of videos in HICom-248K.

Categories defined with natural language
A video about cooking activity.
A video about writing activity.
A video about travel.
A video about sight-seeing activity.
A life record video about exercise.
A life record video about daily life.
A life record video about handcraft.
A life record video about food.
A TV news report video.
A video about computer games.
A video about sports.
A video about football.
A video about basketball.
A video about pets and animals.
A video about action movie scene.
A video about comedy movie scene.
A video about sci-fi movie scene.
A video about crime movie scene.
A video about horror movie scene.
A video about magic show.
A video about acrobatics.
A documentary or TV show about humanity and history.
A documentary or TV show about biography.
A documentary or TV show about geography.
A documentary or TV show about finance and commerce.
A documentary or TV show about literature and art.
A documentary or TV show about biology and medicine knowledge.
A documentary or TV show about finance and commerce knowledge.
A documentary or TV show about technology knowledge.

Table A3. The ablation study on the group strategy for the local-level compression.

Methods	w/ group	VideoMME w/o sub.				MV-Bench	Ego-Schema
		short	mid	long	all		
Unconditional	✓	36.7	34.4	32	34.4	43.7	42.7
	✗	34.7	31.9	31	32.5	42.9	39.9
Conditional	✓	38.8	36.1	33.1	36.0	44.0	43.2
	✗	36.6	33.7	31.2	33.8	43.6	41.6

content of the video mentioned in the instruction.

We collect the videos from Panda-70M [5] and Ego4D [15]. To ensure the diversity of the video sources, we pre-define 29 categories [10, 72] using natural language, select 1,500 videos for each category based on the similarity score calculated by InternVideo2 [54], and randomly select additional 10,000 videos from the whole Panda-70M and Ego4D datasets. The 29 categories are shown in Tab. A2. Fig. A1 shows some examples of our constructed HICom-248K. We use the open-soured Qwen2-VL-72B-Instruct [52] to generate around three instruction-description pairs for each video. The generated descriptions follow the instructions well and also accurately capture the visual content, which is suitable for conditional pre-training.

Table A4. The ablation study on valid and invalid instruction on VideoMME without subtitles. We manually select 326 samples with invalid instructions and 2374 samples with valid instructions.

Methods		Short	Medium	Long	All
Valid	# Samples	808	816	750	2374
	w/o inj.	34.1	33.5	31.1	32.9
	w/ inj.	36.3	35.5	33.2	35.0
Invalid	# Samples	92	84	150	326
	w/o inj.	63.0	47.6	38.7	47.9
	w/ inj.	63.0	47.6	39.3	48.1

Table A5. Ablation study about the Conditional Pre-training stage (CP for short) and HICom-248K data with different training strategies. We keep the projector of LLaVA-OV/LLaVA-Video (*i.e.*, two layers of MLP, 2×2 spatial pooling) to train a baseline with our ablation data. We report the result on Video-MME without subtitles and EgoSchema.

Traning Strategy	Methods	VideoMME	EgoSchema
2 Stage w/o CP	Baseline	36.1	42.5
	HICom	36.0	41.6
2 Stage mix HICom-248K for SFT	Baseline	36.4	43.3
	HICom	36.2	42.4
3 Stage w/ HICom-248K for CP	Baseline	36.2	43.2
	HICom	**36.6**	**43.5**
3 Stage w/ random 248K MCQA for CP	HICom	34.6	40.8

C. More Experiments

C.1. More Ablation Studies

We implement more ablation studies here to demonstrate the superiority and generalization ability of our HICom.

The group strategy at the local level. We introduce the temporal-spatial inductive bias, group the visual tokens, and conduct the local-level conditional compression within each group, preserving the temporal-spatial structure while highlighting the instruction-relevant visual content. We evaluate this grouping strategy for the local-level compression in Tab. A3. Without the grouping strategy, the performance drops significantly, especially on VideoMME and EgoSchema benchmarks, showing the significance of explicitly maintaining the temporal-spatial structure.

Valid and invalid instruction. We notice that not all instructions can provide effective guidance information for capturing visual information, *e.g.*, the instruction of the caption task. We call them the invalid instruction. To evaluate the performance of our HICom in this situation, we manually select out 326 samples with invalid instructions in the VideoMME benchmark. We list some examples of our selected invalid instruction as follows:

- What is this video mainly about?
- What can be learned from this video?
- Which element doesn't show up in the video?

Figure A2. Some video dialogue examples of HICom in the scene of the animated style.

- In what order were the following mentioned in the video?
- According to the video, which of the following statements is true?
- Which of the following accurately describes the content of the video?

As shown in Tab. A4, we test both the unconditional and conditional compression on 326 invalid instructions and the other 2374 valid instructions of VideoMME without subtitles separately. For valid instructions, the conditional compression (w/ inj.) gains 2.1% compared with unconditional compression (w/o inj.). When it comes to invalid instructions, the conditional compression keeps the same results

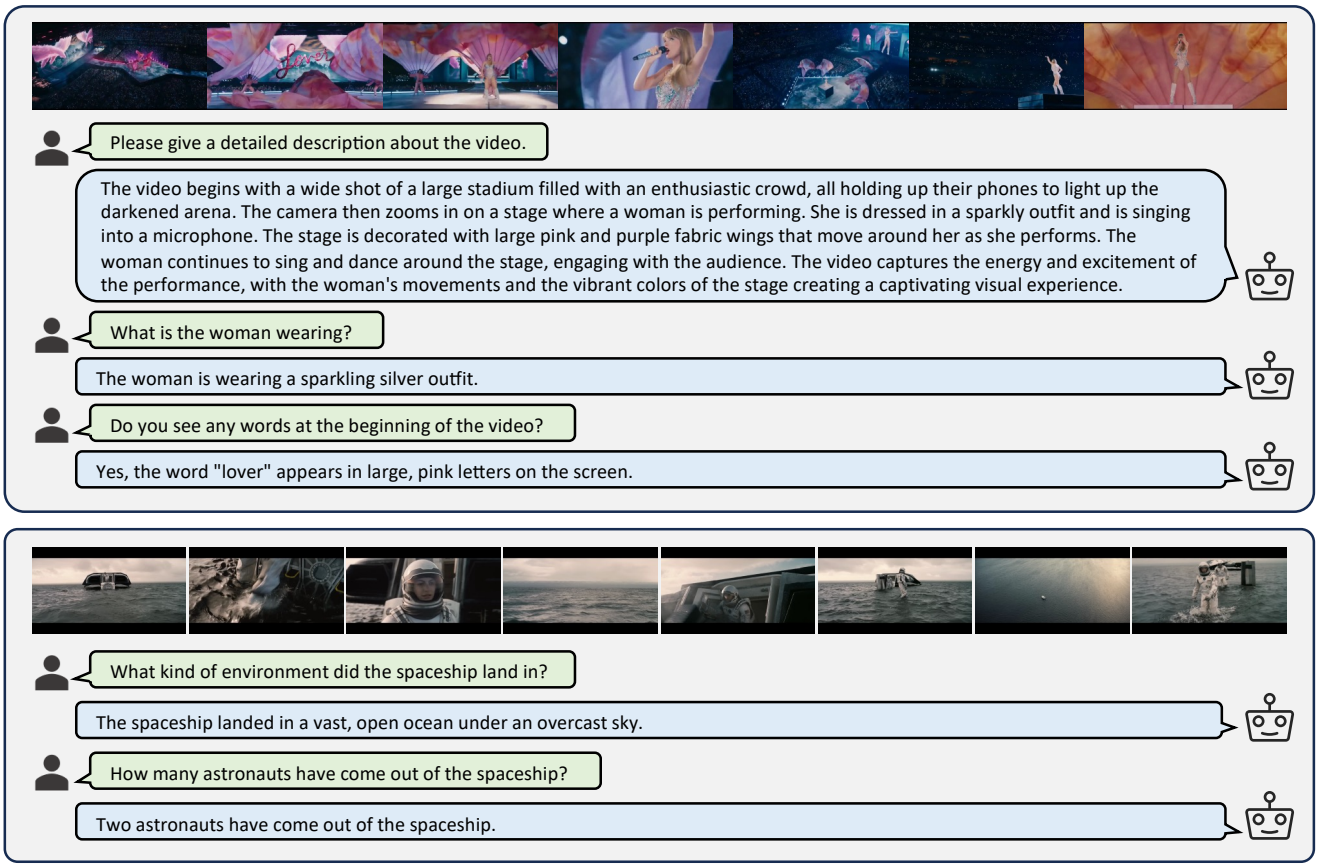


Figure A3. Some video dialogue examples of HICom in the scene of the realistic style.

Table A6. Inference efficiency comparison between LLaVA-OneVision-7B and our HICom-7B. We report the number of parameters, the inference time of each component, and the final throughput.

	Methods	Frames	Vision Encoder	Compressor	LLM	All
Params	LLaVA-OV-7B	32	413M	16M	7.6B	8.0B
	HICom-7B	32	428M	450M+63M	7.6B	8.5B
Time (ms)	LLaVA-OV-7B	32	11.1	2.3	553.7	567.1
	HICom-7B	32	11.1	23.9	102.7	137.7
Throughput (s/video)	LLaVA-OV-7B	32	-	-	-	4.25
	HICom-7B	32	-	-	-	1.51
	HICom-7B	64	-	-	-	1.89
	HICom-7B	128	-	-	-	2.68

as unconditional compression on short and medium videos, and even gains slightly on long videos. We also find the performance of this situation might be easier than valid instruction, as both models perform much better. Thanks to our design of the local-level compression and the group strategy, we argue that the conditional compression will also focus on the global content of the video within each group, and degrade to the situation of unconditional compression, as there also exists this kind of data during training. The conditional compression will not perform lower than the unconditional compression.

Table A7. The comparison of SOTA methods and HICom on MLVU$_{dev}$ benchmark. * indicates we reproduce the results ourselves using the official checkpoint and inference code provided by authors. § donates we inference with a new length of frames trained by sampling 32 frames.

Methods	LLM Size	Frames	Tokens	MLVU$_{dev}$
Video-LLaVA [30]	7B	8	2048	47.3
LLaMA-VID [28]	7B	1fps	2tps	33.2
LongVA [69]	7B	128	18432	56.3
VideoLLaMA2 [7]	7B	16	1152	48.5
LLaVA-OneVision [20]	7B	32	6272	65.3*
LLaVA-Video [72]	7B	32	6272	<u>66.7</u>*
HICom (Ours)	7B	32	680	62.8
HICom (Ours)§	7B	64	1328	65.1
HICom (Ours)§	7B	128	2624	**67.2**

Conditional pre-training stage and HICom-248K data. We further conduct ablation studies on our proposed conditional pre-training stage and HICom-248K data. We use four different training strategy settings, and report their re-

sults of both baseline and our HICom, as shown inTab. A5. We keep the projector of LLaVA-OV/LLaVA-Video (*i.e.*, two layers of MLP, 2×2 spatial pooling) to train a baseline with our ablation data. We find the increase from constructed data for baseline (*i.e.*, the comparison between the first strategy and the third strategy) is not as significant as HICom (averagely 0.4% vs 1.25%). We also find that for HICom, mixing SFT (the second strategy) gains slightly (0.5%) on 2-stage training (the first strategy), but our 3-stage training (the third strategy) outperforms 2-stage training (the first strategy) obviously (1.25%). These two findings demonstrate that our improvement comes more from the additional pre-training strategy, rather than the constructed data themselves. The fourth strategy demonstrates the significance of the data type of the conditional stage, as the performance significantly drops when we change our instruction-followed descriptions to multi-choice QA data, which may confuse the alignment of instruction injection.

Inference efficiency. Apart from the number of visual tokens that are sent into LLM, we also report the throughput to further demonstrate the inference efficiency. The comparison between LLaVA-OneVision-7B and HICom-7B is shown in Tab. A6. We report the time using the same sample, and we only report the LLM time of the first generated token for fair comparison. We report the average time-consuming result of inferring 100 samples as throughput, the time in throughput is larger than our reported time because the throughput counts the time of video loading, the `prepare_inputs_labels_for_multimodal` process, and the LLM generates more than one token. Compared with LLaVA-OneVision, our vision encoder includes an additional projector and therefore contains additional 16M parameters, and the compressor includes an additional 450M text encoder. This leads to our 23.9ms consumption of the compressor with the text encoding process, 21.6ms more than LLaVA-OneVision. However, our compressor significantly reduces the visual tokens, resulting much shorter time consumption of LLM (102.7ms *vs* 553.7ms), as the LLM inference time usually occupies the main part. Therefore, the number of visual tokens can also effectively and accurately reflect the inference efficiency. Thanks to our compression, our final throughput is much faster than LLaVA-OneVision, as our HICom with 128 frames still infers 1.6x faster than LLaVA-OneVision with only 32 frames.

C.2. More Benchmarks

We also report our HICom on MLVU [73] in Tab. A7 as MLVU is a long video benchmark. We report the performance of MLVU's dev split on multi-choice tasks. Our HICom also achieves comparable performance among SOTA 7B models.

C.3. More Qualitative Analysis

We provide some examples of video dialogues in this section to intuitively show the video understanding ability. Fig. A2 shows the situation of animated style videos. Thanks to the design of the local-level compression, our HICom also obtains powerful caption capabilities and gives detailed and accurate captions for videos. Meanwhile, HICom can make judgments based on the understanding of the video. For example, it captures the girl's outfit to draw the conclusion that the girl is cute in the first example video, and think the dancing is beautiful and captivating after watching the second example video. Fig. A3 shows the scene of realistic style videos, which is also easy for HICom to handle. In the first example video, HICom accurately describes the concert scene, and also recognize the word "Lover" on the large screen. In the second movie clip example, HICom can perceive the environment information and accurately count how many people are in the scene.